Parallel 3D FDTD Simulator for Photonic Crystals

Jason S. Ayubi-Moak, Stephen M. Goodnick, Gil Speyer, and Daniel C. Stanzione
Arizona State University, Tempe, AZ
{jason.ayubi-moak, stephen.goodnick, dstanzi, speyer}@asu.edu.

Paul Sotirelis
High Performance Technologies, Inc., Wright-Patterson AFB, Dayton, OH
sotireli@asc.hpc.mil

Abstract

Photonic crystals have shown a great deal of promise for the realization of true integrated optics. Waveguides with small bends may be formed allowing compact integrated photonic circuits to be formed. Full three-dimensional (3D) photonic simulations are required in order to realize very low loss, integrated photonic crystal circuits. Needless to say, the design and fabrication of such fully 3D structures is challenging, and thus efficient simulation tools are necessary to identify the optimum structures for different applications. Researchers at the Department of Defense (DoD) and Arizona State University (ASU) have independently developed parallel Finite Difference Time Domain (FDTD) codes, with the goal of scaling up each simulator for complicated structures such as 3D optical integrated circuits (OIC). As the name implies, FDTD is a popular time-domain method for solving Maxwell's equations for the electric and magnetic fields. These two curl equations are solved explicitly in time over half-step intervals, where the values of one set of field values (e.g., electric fields) are used at the successive interval to solve for the other field (e.g., magnetic field) in a time marching fashion. The goal of our current work is to realize a fully parallel FDTD code scalable to 10^8 FDTD grid points in order to have sufficient resolution to model even a relatively limited number of periods of a given waveguide structure. This requires both a scalable parallel FDTD code, as well as one with the proper boundary conditions and more efficient algorithms to reduce run. The work and results discussed herein address both the scalability and the efficiency of the time-domain algorithm.

Index Terms: Finite-difference time-domain (FDTD) methods, photonic crystals, parallel processing, electromagnetic analysis.

1. Introduction

Photonic crystals or photonic bandgap materials (PBM's) for nanophotonics are currently of great interest for integrated optics supporting future communication systems, surveillance, sensing, and other DoD applications. Such *crystals* are composed of one, two and three-dimensional periodic arrays of scatterers that open gaps in the optical spectrum, and thus allowing filtering and waveguiding at dimensions well below the characteristic wavelengths of the component waves in free space. The inclusions of *defects* in the crystal allow further tuning of the transmission properties of the photonic crystal, permitting selected wavelengths in a narrow spectral range to pass. In order to accurately model not only the photonic crystal, but the coupling into the PBM, time-domain electromagnetics solvers are desired to understand and control the propagation characteristics of electromagnetic radiation through photonic crystal devices. Problem sizes up to 10M grid points are desired by researchers at Space & Naval Warfare Systems Command (SPAWAR) investigating such materials, which necessitates the use of HPC resources, and the development of massively parallel and computationally efficient algorithms for such large scale simulations.

In this work, the development of a 3D parallel FDTD modeling tool is presented in support of this computational need at SPAWAR. While there are numerous options for electromagnetic problems (time-domain, frequency-domain, finite-difference, finite-element, etc.), FDTD is particularly widespread in use for time-domain electromagnetic (EM) simulation due to its relative ease of implementation, its well-established convergence properties, and the well-developed implementation of absorbing boundary conditions. The remaining sections of this paper are organized as follows. In Section 2, the FDTD technique is discussed in more detail as well as the implementation of the absorbing boundary conditions utilized. In addition, a brief

0-7695-3088-5/07 $25.00 © 2007 IEEE

overview of the parallelization methodology used will also be provided in this section. Scalability and speedup comparisons of both existing SPAWAR codes and the newly developed simulator herein will be given in Section 4. Finally, concluding remarks are offered in Section 5.

2. Methodology

A. Finite Difference Time Domain Method

The propagation of EM waves in any medium is completely described by the solution of Maxwell's equations. These vector wave equations can be easily discretized and solved on a grid mesh using the well-known FDTD method[1]. In an isotropic medium, the time-dependant equations, in curl form, and the corresponding constitutive relationships can be written as

$$\frac{\partial \vec{B}}{\partial t} + \nabla \times \vec{E} = 0, \tag{1}$$

$$-\frac{\partial \vec{D}}{\partial t} + \nabla \times \vec{H} = \vec{J}, \tag{2}$$

$$\vec{B} = \mu \vec{H}, \tag{3}$$

$$\vec{D} = \varepsilon \vec{E}, \tag{4}$$

where $\vec{E}$ and $\vec{H}$ represent the three-dimensional electric and magnetic field vectors. The terms $\vec{J}$, μ, and ε represent the three-dimensional current density vector, magnetic permeability, and dielectric permittivity, respectively. In general, both the permittivity and the permeability are each a function of both spatial coordinates and time. Within this work, however, the use of frequency dependant media has not yet been included into the simulation model. Rather, we assume the dielectric and permeability functions to be uniformly defined static quantities on the simulation grid.

By expanding the curl operator in Eqs. (1 and 2) and equating the respective vector components on both sides of the equations, these two equations can be expressed (taking into account the aforementioned assumptions) as a system of six scalar equations

$$-\mu \frac{\partial H_x}{\partial t} = \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \qquad \varepsilon \frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - J_x, \tag{5}$$

$$-\mu \frac{\partial H_y}{\partial t} = \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \qquad \varepsilon \frac{\partial E_y}{\partial t} = \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - J_y, \tag{6}$$

$$-\mu \frac{\partial H_z}{\partial t} = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \qquad \varepsilon \frac{\partial Ez}{\partial t} = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - J_z. \tag{7}$$

where the corresponding permittivity and permeability have been assumed to be time-invariant. Using a centered-difference scheme approximation, each of these six expressions can be discretized and written in an appropriate finite-difference form necessary for updating each component of the electric and magnetic field. For example, the two expressions in Eq. 1.5) above can be expressed as the following set of discretized equations

$$\begin{aligned} -\mu\left(i, j+\tfrac{1}{2}, k+\tfrac{1}{2}\right) & \frac{H_x^{t+\Delta t}\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right) - H_x^{t-\Delta t}\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right)}{\Delta t} \\ &= \frac{E_z^t\left(i, j+1, k+\frac{1}{2}\right) - E_z^t\left(i, j, k+\frac{1}{2}\right)}{\Delta y} \\ &\quad - \frac{E_y^t\left(i, j+\frac{1}{2}, k+1\right) - E_y^t\left(i, j+\frac{1}{2}, k\right)}{\Delta z} \end{aligned} \tag{8}$$

$$\begin{aligned} -\varepsilon\left(i+\tfrac{1}{2}, j, k\right) & \frac{E_x^{t+\Delta t}\left(i+\frac{1}{2}, j, k\right) - E_x^{t-\Delta t}\left(i+\frac{1}{2}, j, k\right)}{\Delta t} \\ &= \frac{H_z^t\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right) - H_z^t\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)}{\Delta y} \\ &\quad - \frac{H_y^t\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) - H_y^t\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right)}{\Delta z} - J_x \end{aligned} \tag{9}$$

which when solved, yield the current value of the x-directed component of both magnetic and electric fields. These expressions treat the field components as spatially separated quantities that are interwoven in both space and time. The method for solving this system of equations can be visualized as a "leapfrog" technique in which the field components are solved for and marched forward in time one after another, and is the underlying mechanism enabling non-dissipative field propagation in this technique. In 3D, the electric and magnetic field components can be visualized as existing on separate but interlaced grids occupying a single cubic unit cell in space. This space lattice is often referred to as a "Yee cell", shown in Figure 1, and illustrates the idea that the discretized domain space is filled with an interwoven array of field contours, representative of the differential and integral forms of Maxwell's equations. Another way to visualize the grid is by realizing that each electric field component is surrounded by four magnetic field components in the plane perpendicular to the specific electric field component or vice-versa in the case of the magnetic field components.

Using either one of the visual representations, the origins for both the electric field and magnetic field coincide with the origin at (0, 0, 0,) on the coordinate axes. The color distinction between electric and magnetic fields in Figure 1 is adopted to highlight the fact that these field components exists on two separate, but orthogonal grids. During numerical simulations, the individual field components are stored in large data arrays whose values

are designated by whole integers, not half integer values as indicated in Eqs. (8 and 9), which are for illustration purposes only. Thus the programming implementation of the individual field update equations is referenced to the (0, 0, 0) as are the half integer values as well. This is done for simplicity in the computations. However, it is important to point out that the fields are still updated as if they were spatially separated as illustrated in Figure 2.

B. Absorbing Boundary Conditions

Implementation and use of the FDTD method for device simulations requires a unique form of boundary conditions. Textbook electromagnetic (EM) wave problems are often defined, with open or "unbounded" domains that extend out to infinity in all directions. Obviously, it is computationally impossible to process and store the unlimited amounts of data that would be required in these ideal cases. Therefore, the simulation domain must be *effectively* truncated in such a manner that it is large enough to fully contain the structure of interest and resolve any region of interest surrounding the enclosed device. In addition, the boundaries should allow for the propagation and *attenuation* of outward traveling waves while minimizing reflections. Many type of boundary conditions have been developed over the years including the radiating boundary[2,3], the matched layer[4-6], and the one-way approximation of the wave equation[7] which was later applied to EM wave problems by Mur[8]. Although each of these techniques was relatively effective at absorbing outward traveling waves, they were only applicable to waves traveling perpendicular to the boundary layer itself. Thus, their use imposed the constraint of having to set up numerical boundaries far enough away from the wave source so that the outward waves were plane wave like in nature and perpendicular to the absorbing boundary layer.

The specific approach utilized in this work is an absorbing boundary condition (ABC) developed by Bérenger[9,10] in 1994 which he called the *perfectly matched layer* (PML). This approach also utilizes an absorbing outer boundary layer but with a specially designed matching medium designed to absorb propagating EM waves. The original formulation of this technique (there have been several variants of it recently in the literature) involves a novel *field-splitting technique* within the boundary layer cells which results in an artificial or imaginary layer that can *theoretically* absorb any kind of traveling wave at any frequency, regardless of the direction of wave travel, and without reflection. Thus, the use of the PML "simulates" the physical propagation of waves out to infinity while effectively reducing the negative and unwanted effect of reflections back into the computational space. Implementation of the PML is a *conceptually* straightforward task that involves the application of unique, non-zero electric and magnetic conductivities within a fixed number of boundary layers surrounding the simulated device. These conductivities are generally graded as one moves outward toward to outmost boundary cell. To include the PML boundary condition, the expressions given in Eqs. (5–7) are reformulated and expanded into a set of twelve new update equations,

$$\varepsilon\frac{\partial E_{xy}}{\partial t}+\sigma_y E_{xy}=\frac{\partial(H_{zx}+H_{zy})}{\partial y}, \qquad \mu\frac{\partial H_{xy}}{\partial t}+\sigma_y^* H_{xy}=-\frac{\partial(E_{zx}+E_{zy})}{\partial y}, \quad (10)$$

$$\varepsilon\frac{\partial E_{xz}}{\partial t}+\sigma_z E_{xz}=-\frac{\partial(H_{yz}+H_{yx})}{\partial z}, \qquad \mu\frac{\partial H_{xz}}{\partial t}+\sigma_z^* H_{xz}=\frac{\partial(E_{yz}+E_{yx})}{\partial z}, \quad (11)$$

$$\varepsilon\frac{\partial E_{yz}}{\partial t}+\sigma_z E_{yz}=\frac{\partial(H_{xy}+H_{xz})}{\partial z}, \qquad \mu\frac{\partial H_{yz}}{\partial t}+\sigma_z^* H_{yz}=\frac{\partial(E_{xy}+E_{xz})}{\partial z}, \quad (12)$$

$$\varepsilon\frac{\partial E_{yx}}{\partial t}+\sigma_x E_{yx}=\frac{\partial(H_{zx}+H_{zy})}{\partial x}, \qquad \mu\frac{\partial H_{yx}}{\partial t}+\sigma_x^* H_{yx}=\frac{\partial(E_{zx}+E_{zy})}{\partial x}, \quad (13)$$

$$\varepsilon\frac{\partial E_{zx}}{\partial t}+\sigma_x E_{zx}=\frac{\partial(H_{yz}+H_{yx})}{\partial x}, \qquad \mu\frac{\partial H_{zx}}{\partial t}+\sigma_x^* H_{zx}=-\frac{\partial(E_{yz}+E_{yx})}{\partial x}, \quad (14)$$

$$\varepsilon\frac{\partial E_{zy}}{\partial t}+\sigma_y E_{zy}=-\frac{\partial(H_{xy}+H_{xz})}{\partial y}, \qquad \mu\frac{\partial H_{zy}}{\partial t}+\sigma_y^* H_{zy}=\frac{\partial(E_{xy}+E_{xz})}{\partial y}, \quad (15)$$

in which each field component has been split up into a summation of two terms (e.g., $E_x = E_{xy} + E_{xz}$) and σ_x, σ_y, σ_z, σ_x, σ_y, and σ_z, represent the electric and magnetic (*) conductivities within the boundary layers. Note here that, if $\sigma_x = \sigma_y = \sigma_z = 0$, and $\sigma_x = \sigma_y = \sigma_z = 0$, then Eqs. (1.10–1.15) simply reduce down to the form of Maxwell's equation appropriate in a vacuum. On the other hand, if $\sigma_x = \sigma_y = \sigma_z$ and $\sigma_x = \sigma_y = \sigma_z = 0$, then Eqs. (10–15) reduce to the equations of a conductive medium. Finally, if $\sigma_x = \sigma_y = \sigma_z = 0$ and $\sigma_x^* = \sigma_y^* = \sigma_z^*$, then Eqs. (10–15) reduce to the equations of an absorbing medium.

As waves travel outward and into the PML layer, these conductivities (which can be either linearly or polynomially graded) effectively attenuate the wave amplitude absorbing its energy into this artificial layer. The outermost layer of grid cells is defined as a perfect electric conductor (PEC) by setting the tangential components of the electric field to zero and reflect any remaining waves backwards through the PML layer further absorbing any remaining or lingering waveforms.

The beauty of this technique lies in the fact that the FDTD grid itself is virtually unchanged. The only change comes from the additional six field components required within the PML layer. Within the inner simulation domain, the discretized form of Eqs. (5–7) are solved. In the PML layer, however, the discretized form of Eqs. (10-15) are utilized. Thus, within the absorbing layer there are 12 subcomponents of the electric and magnetic fields that must be accounted for. Figure 2 illustrates the implementation of a PML layer surrounding a 3D computational domain. The subscripting in the figure is included for the most general case in which the specific

PML layer thicknesses at each boundary are arbitrary. In other words, the front face (i.e., front y-z plane) might be composed of a certain number of layers while the back face may contain a completely different number of absorbing layers. Thus, the specific component conductivities in each of these regions can be completely unique and is accounted for via the integer subscripting in the coordinates given in the figure.

C. CFL Stability Criterion

Another crucial challenge faced when explicitly solving Maxwell's equations on a finite grid involves the maximum timestep over which the resulting finite-difference expressions can be resolved. This upper bound on the timestep is imposed due to both the temporal and spatial discretization required in the finite-differencing of Eqs. (15–17). This inherent stability limit, know as the Courant-Frederichs-Levy (CFL) criterion severely limits the simulation timestep to approximately 10^{-17}s for precisely the sub-micron scale dimensions required in a typical photonic crystal simulations. In 3D, this condition is expressed as

$$\Delta t \leq \frac{1}{c\sqrt{\left(\frac{1}{\Delta x}\right)^2+\left(\frac{1}{\Delta y}\right)^2+\left(\frac{1}{\Delta z}\right)^2}}, \qquad (16)$$

where c represents the highest wave velocity across the simulated domain, Δt is the timestep, and Δx , Δy , and Δz are the spatial dimensions of each grid cell. Choosing a timestep lower than the prescribed upper bound is necessary to ensure numerical instability (i.e., suppression of non-physical oscillations and unbounded field growth) when using an explicit FDTD approach.

The CFL limitation can be overcome by using a new alternate-direction implicit (ADI) approach that has been recently reported[12,13]. This new and more complex implicit formulation of the discretized Maxwell's equations, called the ADI-FDTD method, results in a relaxation of the CFL criterion and allows EM fields to be updated using timesteps several orders of magnitude greater than that dictated by the conventional limit given in Eq. (16). Although this new approach does offer the possibility of significant reductions in the overall device simulation time required, it does so at the cost of increased computational overhead per timestep and reduced accuracy in the resulting field magnitudes computed. The use of larger timesteps has been linked to increases in both local and global error across the simulation domain resulting from the build up of truncation errors in the field updates[14]. However, these values are typically small and generally acceptable if the method is utilized within the regime where the simulation grid is overspecified due to physical properties (i.e., doping, critical component dimensions, etc.) but is not necessary to adequately resolve the smallest wavelength of interest (i.e., highest frequency of interest). Furthermore, the loss of accuracy is dependant to a large degree upon the specific device geometry and the total time simulated.

D. Domain Decomposition and Parallelization

A code implementing the methods described has been constructed using the Message Passing Interface (MPI). In order to reduce the number of communication steps, redundant calculations at boundary regions were employed similar to that in Reference 15. In this way, only E information needs to be transmitted as H data is current in all subdomains. Care was taken to preserve scalability in the code by dynamically allocating to each MPI task only the amount of the grid required to solve the problem. A domain decomposition has been implemented in one- (1D) and three-dimensions within the Message Passing Interface (MPI) framework of our simulator as shown in Figure 3. The initial 1D domain decomposition resulted in good scaling for crystal geometries with a long thin channel. The 3D decomposition, however, seemed to work best for more general geometries, and particularly for larger problems. This decomposition is the default in the code delivered to the DoD user community.

3. Results

The software development portion of this task was divided into two components, implementation of the method described above in a new, scalable parallel code, and comparison of this new code to an existing code implementing an FDTD method known as "PNEWS". A comparison of the speedup between both codes is given in Figure 4. In the analysis of the PNEWS code, scaling was found to be limited by a number of shared-memory assumptions in the code (e.g., the entire problem grid was allocated on each task). The initial analysis was performed on distributed memory cluster systems, which made a fair comparison of the two codes difficult, as the PNEWS code would only succeed for relatively small problem sizes, hence the low processor counts in Figure 4. Even at these problem sizes, the new implementation of the FDTD3D code shows better scaling performance.

Analysis of the new code has continued on much larger problem sizes and on a variety of systems, both at Arizona State University and at the DoD Major Shared Resource Centers (MSRCs). To date, scale up of the code to more than 10^8 points with linear speedups on several hundred processors has been achieved. The code as structured appears to scale equally well on both cluster

architectures and shared memory systems, such as the SGI Altix system at the Aeronautical Systems Center MSRC. Figure 5, shows a visualization of the execution performance of a 60 processor job, taken with the analysis tool "Jumpshot". In the figure, horizontal lines map to the execution of individual MPI Tasks. Wherever a colored region is shown on the line, the task is in the midst of an MPI communication operation. Areas without shading indicate computation occurring. In general, the less shaded regions, the more efficient the overall execution time will be. In this instance, the figure shows that each iteration of the algorithm is dominated by computation time rather than communication time, which is a positive sign. Figure 6 shows speedup results for three different geometries on 32, 64, 128, and 256 processors, for the final code with the 3D domain decomposition. This graph was produced using a Linux cluster with Intel EM64T quad-core processors and an Infiniband interconnection network. The legend indicates the size of the total grid in the x, y, and z dimensions. The x and z sizes were held constant, and the number of elements in the y dimension was varied. For the largest problem size, the 32 and 64 processor runs were omitted, as the system lacked the memory to run a problem of that size in core on such a small number of processors. In all cases, scaling appears excellent. There is some flattening of the speedup curve in the case of the smallest problem as the number of processors increases, as expected, but large problems seem to scale well to large systems.

4. Conclusions

This paper describes the application of the 3D FDTD method to a photonic crystal. Parallelization enabled an efficient computation over the large crystal domain to be performed. Performance results indicate that this calculation can be performed over the hundreds of processors necessary for such a large domain and for the large number of time steps that will be needed. In future work, we will explore the issues involved with using a very large number of processors with this problem.

Acknowledgments

The authors would also like to thank Dr. Scott Rodgers at SPAWAR for many helpful discussions.

References

1. Yee, K., "Numerical solution of initial boundary value problem involving Maxwell's equations in isotropic media." *IEEE Trans. Antennas and Propagat.*, vol. 14, 302, 1966.

2. Merewether, D., "Transient Currents Induced on a Metallic Body of Revolution by an Electromagnetic Pulse." *IEEE Trans. Electromagn. Compat.*, vol. 13, pp.41–44, 1971.

3. Holland, R., "THREDE: A free-field EMP coupling and scattering code (electromagnetic scattering from aircraft)." *IEEE Trans Nucl. Sci.*, vol. NS-24, pp. 2416–2421, Dec. 1971.

4. Bérenger, J.P., Note Technique DGA/ETCA/DET/390, 1977, (unpublished).

5. Bérenger, J.P, in *Actes du Colloque CEM, CNFRS-URSI, Trégastel, France*, 1983.

6. Holland, R. and J.W. Williams, "Total-field versus scattered-field finite-difference codes: A comparative assessment." *IEEE Trans. Nucl. Sci.*, vol. NS-30, pp. 4583, 1983.

7. Engquist, B. and A. Majda, "Absorbing Boundary Conditions for the Numerical Simulation of Waves." *Math. Comput.*, vol. 31, no. 139, pp. 629–651, July 1977.

8. Mur, G., "Absorbing boundary conditions for the finite difference approximation of the time domain electromagnetic field equations." *IEEE Trans. Electromagn. Compat.*, vol. EMC-23, pp. 377–382, Nov. 1981.

9. Bérenger, J.P., "Perfectly matched layer for the FDTD solution of wave-structure interaction problems." *IEEE Trans. Antennas Propagat.*, vol 44, pp. 110–117, Jan. 1994.

10. Bérenger, J.P., "Three-dimensional perfectly matched layer for the absorption of electromagnetic waves" *J. Comput. Phys.*, vol. 127, pp. 363–379, 1996.

11. Courant, R., K. Frederichs, and H. Levy, "On the partial difference equations of mathematical physics." *IBM Journal*, 215, 1967.

12. Namiki, T., "A New FDTD Algorithm Based on Alternating-Direction Implicit Method." *IEEE MTT*, 47(10), p. 2003, 1999.

13. Zheng, F., *et al.*, "A Finite-Difference Time-Domain Method Without the Courant Stability Conditions." *IEEE Microwave Guided Wave Lett.*, 9(11), p. 441, 1999.

14. Liu, G., and S.D. Gedney, "Perfectly matched layer media for an unconditionally stable three-dimensional ADI-FDTD method." *IEEE Microwave Guided Wave Lett*, 10(7), p. 261, 2000.

15. Hanawa, T. and S. Ikuno, "Large-Scale Simulation for Optical Propagationin 3-D Photonic Crystal Using the FDTD Method With Parallel Processing." *IEEE Transactions on Magnetics*, 43(4), p. 1545, 2007.

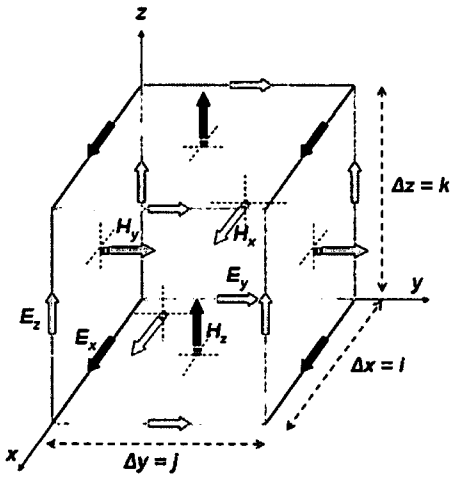

Figure 1. 3D illustration of interlaced electric and magnetic field components in the Yee cell

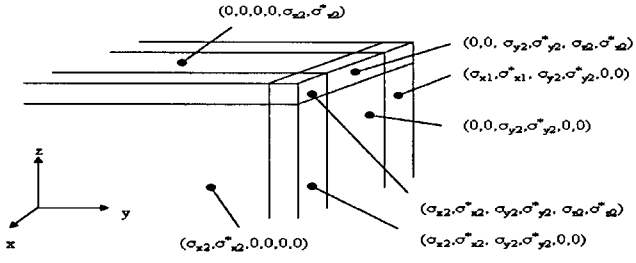

Figure 2. Implementation of PML surrounding 3D simulation domain

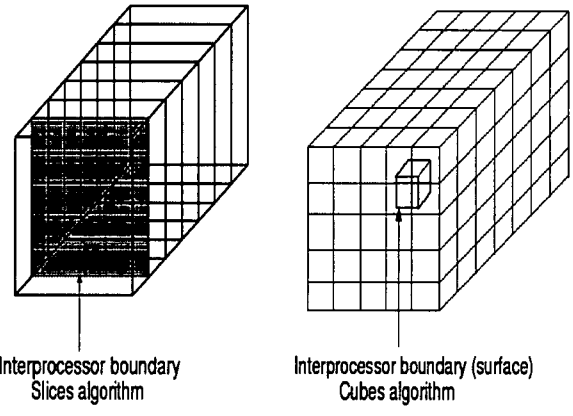

Figure 3. Simple domain decomposition of FDTD domain into "slices" and "cubes." A dedicated processor is assigned to each domain slice or cube and communications are transmitted across interfaces.

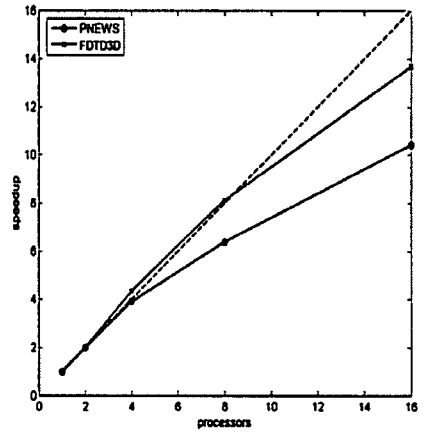

Figure 4, Comparison of speedup versus number of processors for ASU code (FDTD3D) and DoD code (PNEWS). Data taken from simulation runs performed on ASU HPC systems. The dotted line represents the ideal linear speedup.

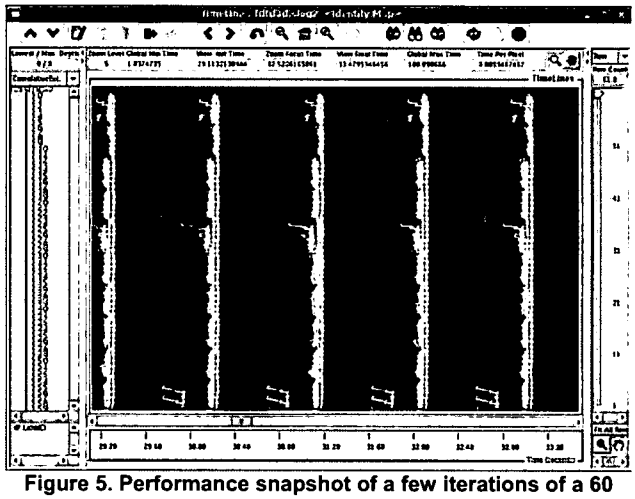

Figure 5. Performance snapshot of a few iterations of a 60 processor run of the FDTD program run on a 2×6×5 grid. Each horizontal line corresponds to a processor. Shaded (green) areas indicate inter-processor communications.

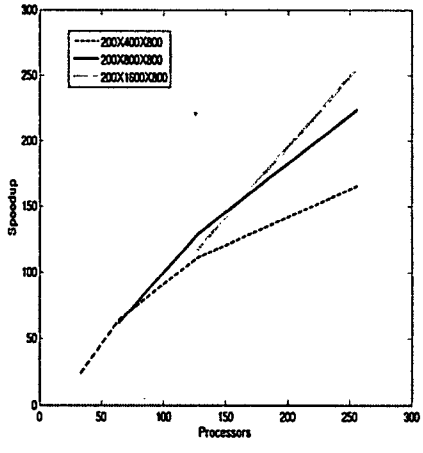

Figure 6. Speedup plots on increasingly larger domains